A Comparison of Naïve Bayes Algorithms and Support Vector Machines Kernels In Spam Classification

Rita McCue
University of California, Santa Cruz
12/7/09
1 Introduction

2 Naïve Bayes Algorithms

3 Support Vector Machines and SVMLib

4 Comparative Results

5 Conclusions

6 Further References
1 Introduction
2 Naïve Bayes Algorithms
3 Support Vector Machines and SVMLib
4 Comparative Results
5 Conclusions
6 Further References
Email classification using the Spam/Ham dataset from homework 3

- Boolean Spam/Ham label and 2000 “Features”
- All features negated and concatenated on the same email row, resulting in 4000 features per email
- 1500 Training emails (1021 Ham and 479 Spam)
- 500 Testing Emails
- 5–fold Cross–Validation (SVM only) performed on Training data, parameters tuned and the best chosen for testing
1 Introduction

2 Naïve Bayes Algorithms

3 Support Vector Machines and SVMLib

4 Comparative Results

5 Conclusions

6 Further References
Naive Bayes Algorithms

- A Naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.

- Comparatively easy to implement.

- Several alternative Naïve Bayes algorithms exist, but all seem to result in the same level of accuracy as the basic NB classifier; the alternate algorithms value is in the fact that they sometimes allow "the same level of accuracy to be reached with fewer attributes."
Naïve Bayes Algorithms (Cont)

- Basic Naïve Bayes

\[
\frac{p(c_s) \cdot p(\vec{x}|c_s)}{p(c_s) \cdot p(\vec{x}|c) + p(c_h) \cdot p(\vec{x}|c_h)} > T
\]

- Multi–Variate Bernoulli Naïve Bayes

\[
\frac{p(c_s) \cdot \prod_{i=1}^{m} p(t_i|c_s)^{x_i} \cdot (1 - p(t_i|c_s))^{1-x_i}}{\sum_{c \in \{c_s,c_h\}} p(c) \cdot \prod_{i=1}^{m} p(t_i|c)^{x_i} \cdot (1 - p(t_i|c))^{1-x_i}} > T,
\]

where \( g(x_i; \mu_{i,c}, \sigma_{i,c}) = \frac{1}{\sigma_{i,c} \sqrt{2\pi}} \cdot e^{-\frac{(x_i - \mu_{i,c})^2}{2\sigma_{i,c}^2}} \)

- Multi–Variate Gauss Naïve Bayes

(not designed for boolean features, implemented anyway)
Naïve Bayes Algorithms (Cont)

- Multinomial Naïve Bayes, Boolean Attributes
  \[ p(c) \cdot \prod_{i=1}^{m} p(t_i|c)^{x_i} \]
  \[ \sum_{c \in \{c_1, c_2\}} p(c) \cdot \prod_{i=1}^{m} p(t_i|c)^{x_i} > T \]
  \[ p(t|c) = \frac{1 + N_{t,c}}{m + N_c} \]
  \[ N_c = \sum_{i=1}^{m} N(t_i, c) \]

- Multinomial Naïve Bayes, Term Freq Attributes
  (identical, but requiring a different data set and thus not implemented)

- Flexible Bayes
  (also requiring a different data set and thus not implemented)

\[ p(x_i|c) = \frac{1}{L_{i,c}} \cdot \sum_{l=1}^{L_{i,c}} g(x_i; \mu_{i,c,l}, \sigma_{i,c}) > T \]
1 Introduction
2 Naïve Bayes Algorithms
3 Support Vector Machines and SVMLib
4 Comparative Results
5 Conclusions
6 Further References
A Support Vector Machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space

Kernel Trick – solves non-linear problems by mapping the original non-linear observations into a higher-dimensional space, and then using a linear classifier.

- Any continuous, symmetric, positive semi-definite kernel function can be expressed as a dot product when moved to a higher-dimensional space. (Mercer’s Theorem)
- Wherever a dot product is used, it is replaced with the kernel function
Support Vector Machines and SVMLib

Kernels available in the SVMLib implementation

- **Linear Kernel**
  \[ K(x_i, x_j) = x_i^T x_j \]

- **Polynomial Kernel**
  \[ K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \]

- **Radial Basis Function Kernel**
  \[ K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma > 0 \]

- **Sigmoid Kernel**
  \[ K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \]
Support Vector Machines and SVMLib

- 5-Fold Cross Validation done on training data to tune parameters $c$ and $\gamma$

- Default $c=1$, $\gamma=1/\text{NumFeatures}$

- 5 distinct $c$ and $\gamma$ options were tested
  - $c = [.001, .01, .1, 1, 10]$;
  - $\gamma = [.0000625, .000125, .00025, .0005, .001]$;
1 Introduction

2 Naïve Bayes Algorithms

3 Support Vector Machines and SVMLib

4 Comparative Results

5 Conclusions

6 Further References
Comparative Results

- Naïve Bayes Algorithms
Comparative Results

- Support Vector Machine Algorithms

(Using Default Tuning Parameters $c=1$, $\gamma=1/\text{features}$)
Comparative Results

- Support Vector Machine Algorithms

(After 5-fold Cross validation, Resulting in different Parameter Choices for Each kernel)
Comparative Results (Testing Data)

<table>
<thead>
<tr>
<th>Naïve Bayes Alg</th>
<th>Results/Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Naïve Bayes</td>
<td>489/500, 98%</td>
</tr>
<tr>
<td>Multinomial Naïve Bayes</td>
<td>479/500, 96%</td>
</tr>
<tr>
<td>Multivariate Naïve Bayes</td>
<td>469/500, 94%</td>
</tr>
<tr>
<td>Multivariate Gauss Naïve Bayes</td>
<td>351/500, 70%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SVM Kernels</th>
<th>Results/Accuracy (Pre Tuning)</th>
<th>Results/Accuracy (Post Tuning)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radial Basis Kernel</td>
<td>425/500, 85%</td>
<td>483/500, 97%</td>
</tr>
<tr>
<td>Sigmoid Kernel</td>
<td>366/500, 73%</td>
<td>481/500, 96%</td>
</tr>
<tr>
<td>Linear Kernel</td>
<td>480/500, 96%</td>
<td>479/500, 96%</td>
</tr>
<tr>
<td>Polynomial Kernel</td>
<td>351/500, 68%</td>
<td>351/500, 70%</td>
</tr>
</tbody>
</table>
1 Introduction
2 Naïve Bayes Algorithms
3 Support Vector Machines and SVMLib
4 Comparative Results
5 Conclusions
6 Further References
Conclusions

- Given our dataset, the far simpler Naïve Bayes implementation also performs far better, on average, than the SVM implementation tested (SVMLib)
- However, as SVMLib is a freely available, already implemented library, it does perform almost as well as the Naïve Bayes implementation on this data set and would likely work better on some other data sets. It is also possible that SVMLib, with further turning, would give a better result.
Conclusions

- Further work might involve minimizing the features used by the Naïve Bayes algorithms and comparing the accuracy rates of the different versions

Rita McCue (UCSC). A Comparison of Naïve Bayes and Support Vector Machines
Further References

